南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

课程详述

COURSE SPECIFICATION

以下课程信息可能根据实际授课需要或在课程检讨之后产生变动。如对课程有任何疑问，请联系授课教师。

The course information as follows may be subject to change, either during the session because of unforeseen circumstances, or following review of the course at the end of the session. Queries about the course should be directed to the course instructor.

1.	课程名称 Course Title	**软件测试 Software Testing**
2.	授课院系 Originating Department	计算机科学与工程系 Department of Computer Science and Technology
3.	课程编号 Course Code	CS409
4.	课程学分 Credit Value	3
5.	课程类别 Course Type	专业选修课 Major Elective Courses
6.	授课学期 Semester	秋季 Fall
7.	授课语言 Teaching Language	英文 English
8.	授课教师、所属学系、联系方式（如属团队授课，请列明其他授课教师） Instructor(s), Affiliation& Contact （For team teaching, please list all instructors）	陈馨慧，助理教授，计算机科学与工程系，tansh3@sustech.edu.cn 陈馨慧, Assistant Professor, Department of Computer Science and Engineering, tansh3@sustech.edu.cn
9.	实验员/助教、所属学系、联系方式 Tutor/TA(s), Contact	
10.	选课人数限额(可不填) Maximum Enrolment （Optional）	

11. 授课方式 Delivery Method	讲授 Lectures	习题/辅导/讨论 Tutorials	实验/实习 Lab/Practical	其它(请具体注明) Other（Please specify）	总学时 Total
学时数 Credit Hours	32		32		64

12. 先修课程、其它学习要求 Pre-requisites or Other Academic Requirements	CS304 软件工程 Software Engineering
13. 后续课程、其它学习规划 Courses for which this course is a pre-requisite	
14. 其它要求修读本课程的学系 Cross-listing Dept.	

教学大纲及教学日历 SYLLABUS

15. 教学目标 Course Objectives

This course will introduce systematic and organized approaches to software testing This course will also provide an overview of the concepts of software testing, including unit testing, integration testing and various testing coverage criteria. This course also aims to introduce several program analysis techniques from the perspective of testing. This course also could provide hands-on experience on some well-known research prototypes in software testing.

16. 预达学习成果 Learning Outcomes

1. Students will learn about the several software development and testing terminologies that will be useful for answering common questions in programming interviews.

2. Students will learn how to systematically test a program and fulfil several coverage criteria.

3. Students will learn about advanced program analysis techniques to model software for testing.

4. Students will have the opportunity to try several testing techniques that could automatically generate test inputs during software development.

5. Students will be able to apply their testing knowledge on a group-based software project.

17. 课程内容及教学日历 （如授课语言以英文为主，则课程内容介绍可以用英文；如团队教学或模块教学，教学日历须注明主讲人）
Course Contents (in Parts/Chapters/Sections/Weeks. Please notify name of instructor for course section(s), if this is a team teaching or module course.)

Week	Hours	Description
Week 1: Introduction	2	This class will give a brief overview of the consequences of software errors and the importance of finding bugs earlier in the software development.
Week 2: Example Testing Session	2	In this class, students will have the opportunity to walk through a bug-finding and testing session. Students will also get a brief overview about some commonly used testing terminologies.
Week 3: Unit Testing (JUnit)	2	This class will introduce the concepts of unit testing and the different levels of testing through several examples.
Week 4: Test Driven Development (TDD)	2	This class will introduce a software development approach where students will write specifications as executable test code before writing the complete source code.
Week 5: Statement Coverage	2	This class will introduce several metrics used to access the quality of test cases. The basic metrics include statement coverage and branch coverage.
Week 6: Graph Coverage	2	This class will continue introducing other more completed metrics used to access the quality of test cases. Moreover, students will learn about different representations of software programs that are important in program analysis.
Week 7: Data flow coverage	2	This class will introduce more metrics used to access the quality of test cases.
Week 8: Graph Coverage for Specifications	2	This class will teach students about how to design programs by writing specifications.
Week 9: Logic Testing	2	This class will introduce another way of modeling software by representing programs based on its logics.
Week 10: Logic Coverage	2	This class will introduce the concepts of logic coverage and other coverage criteria, including predicate coverage and clause coverage.
Week 11: Input Space Partitioning	2	This class will introduce an approach to systematically test a program by dividing the input space based on the given types of inputs.
Week 12: Mutation Testing	2	This class will introduce the concepts of mutations and how small program modifications could be used to model potential software bugs.
Week 13: Regression Testing	2	This class will talk about a practical testing approach that is commonly used to ensure that new changes introduced to a software does not break existing functionalities of the program.

Week 14: Continuous Integration	2	This class will introduce the concepts of continuous integration and how continuous integration affects software development and testing.
Week 15: Finding bugs in Java programs with Automated Testing Tools	2	This class will introduce existing approaches in automatically generate test cases for Java programs.
Week 16: Finding bugs in Android apps with Automated Testing Tools	2	This class will introduce existing approaches in automatically generate test cases for Android apps.

Practical (32 hours)

- Setting up eclipse and JUnit (4 hours)
- Practice Test-driven Development (2 hours)
- Code coverage with EclEmma (2 hours)
- Project / assignment (2 hours)
- Project / assignment (2 hours)
- Project / assignment (2 hours)
- Project / assignment (2 hours)
- Project / assignment (2 hours)
- Project / assignment (2 hours)
- Regression Testing (2 hours)
- Test Generation with Randoop (2 hours)
- Test generation with Monkey (2 hours)
- The class project will be a group-based project where students will be divided into groups of size 2/3. The objective of the project is to allow student to apply the knowledge that they learn in the class to testing real-world applications. Students in each group will get to choose from a list of open-source software projects for testing. For each week, students will apply the testing techniques/criteria that they learn from the class in their selected project and write a report on the effectiveness of the employed technique. At the end of the semester, each group will need to submit a final report that compares all techniques applied in each week and summarizes their findings. Below is the tentative schedule for the class project:
- Measuring code coverage on software projects (2 hours)
- Manual Testing on software projects (2 hours)
- Automated Testing on software projects (2 hours)

18. 教材及其它参考资料 Textbook and Supplementary Readings

Textbook:

Introduction to Software Testing by Paul Ammann and Jeff Offutt

Cambridge University Press, February 2008, ISBN-13: 9780521880381

课程评估 ASSESSMENT

19.

评估形式 Type of Assessment	评估时间 Time	占考试总成绩百分比 % of final score	违纪处罚 Penalty	备注 Notes
出勤 Attendance				
课堂表现 Class Performance				
小测验 Quiz				
课程项目 Projects		20%		
平时作业 Assignments		40%		
期中考试 Mid-Term Test				
期末考试 Final Exam		20%		
期末报告 Final Presentation		20%		
其它（可根据需要改写以上评估方式） Others (The above may be modified as necessary)				

20. 记分方式 GRADING SYSTEM

☑ A. 十三级等级制 Letter Grading
☐ B. 二级记分制（通过/不通过） Pass/Fail Grading

课程审批 REVIEW AND APPROVAL

21. 本课程设置已经过以下责任人/委员会审议通过
This Course has been approved by the following person or committee of authority